



HAL
open science

An Optimization Framework for Regularized Linearly Coupled Matrix-Tensor Factorization

Carla Schenker, Jérémy E Cohen, Evrim Acar

► **To cite this version:**

Carla Schenker, Jérémy E Cohen, Evrim Acar. An Optimization Framework for Regularized Linearly Coupled Matrix-Tensor Factorization. EUSIPCO 2020 - 28th European Signal Processing Conference, Jan 2021, Virtual, Netherlands. pp.1-5. hal-03038083

HAL Id: hal-03038083

<https://hal.science/hal-03038083>

Submitted on 3 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Optimization Framework for Regularized Linearly Coupled Matrix-Tensor Factorization

Carla Schenker
*Simula Metropolitan Center for
 Digital Engineering & Oslo Metropolitan Univ.*
 Oslo, Norway
 carla@simula.no

Jeremy E. Cohen
*University of Rennes,
 Inria, CNRS, IRISA*
 Rennes, France
 jeremy.cohen@irisa.fr

Evrin Acar
*Simula Metropolitan Center for
 Digital Engineering*
 Oslo, Norway
 evrim@simula.no

Abstract—An effective way of jointly analyzing data from multiple sources, in other words, data fusion, is to formulate the problem as a coupled matrix and tensor factorization (CMTF) problem. However, one major challenge in data fusion is that due to eclectic characteristics of data stemming from different sources, various constraints and different types of coupling between data sets should be incorporated. In this paper, we propose a flexible and efficient algorithmic framework building onto Alternating Optimization (AO) and Alternating Direction Method of Multipliers (ADMM) for coupled matrix and tensor factorizations incorporating a variety of constraints and coupling with linear transformations. Numerical experiments demonstrate that the proposed approach is accurate, computationally efficient with comparable or better performance than available CMTF methods while being also more flexible.

Index Terms—tensor factorizations, coupled tensor factorizations, linear couplings, AO-ADMM

I. INTRODUCTION

In many areas of science, patterns of interest can be captured by multiple types of measurements from complementary modalities. For instance, brain activity patterns, *e.g.*, temporal and spatial patterns, can be captured using both electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) signals, but with different resolutions. Similarly, in metabolomics, multiple analytical techniques are used to measure chemical compounds in biological samples providing complementary views of underlying biological processes. Joint analysis of data from multiple sources, also called *data fusion*, exploits these complementary measurements and allows for better interpretability and, potentially, more accurate recovery of patterns of interest.

While some datasets are represented as matrices, many take the form of higher-order tensors. Coupled matrix and tensor factorizations (CMTF) have been used for joint analysis of such datasets in many domains including neuroscience [1]–[4], and omics [5]. In such coupled factorizations, each tensor \mathcal{T}_i is modeled by a low-rank approximation. One of the most popular tensor factorization methods, Canonical Polyadic Decomposition (CPD) (also known as CANDECOMP/PARAFAC (CP)) [6]–[8] models a tensor \mathcal{T}_i of order $D_i \geq 2$ as the sum of R_i rank-one components,

$$\mathcal{T}_i \approx \sum_{r=1}^{R_i} \mathbf{C}_{i,1}(:, r) \circ \mathbf{C}_{i,2}(:, r) \circ \dots \circ \mathbf{C}_{i,D_i}(:, r) =: \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i},$$

where $\mathbf{C}_{i,d}(:, r)$ is the r -th column of factor matrix $\mathbf{C}_{i,d}$ and \circ denotes the vector outer product [9]. Columns of factor matrices may reveal patterns of interest. For $D_i = 2$, this formulation is a generic matrix decomposition. Joint factorization problems are often formulated by extracting the same factor matrix from the coupled mode [1], [10], [11], *e.g.*, if a third-order tensor \mathcal{T}_1 is coupled with a matrix \mathbf{T}_2 in the first mode, the optimization problem is given as:

$$\begin{aligned} \min_{\{\mathbf{C}_{i,d}\}_{i=1,2}^{d \leq D_i}} \quad & \|\mathcal{T}_1 - \llbracket \mathbf{C}_{1,1}, \mathbf{C}_{1,2}, \mathbf{C}_{1,3} \rrbracket\|_F^2 + \|\mathbf{T}_2 - \mathbf{C}_{2,1} \mathbf{C}_{2,2}^T\|_F^2 \\ \text{s.t.} \quad & \mathbf{C}_{1,1} = \mathbf{C}_{2,1} \end{aligned}$$

However, factors corresponding to the coupled mode are not necessarily equal in different datasets. Therefore, in this paper, we focus on more general linear coupling relationships. In particular, we cover the cases, where not all factors are shared between tensors [5], or factors are sampled from a continuous phenomenon with different sampling rates [12]. Using such linear couplings, different resolutions of EEG and fMRI signals have been previously incorporated while jointly analyzing data from these two modalities [2], [3].

Furthermore, in many applications, constraints or regularizations on factor matrices are essential to obtain physically meaningful patterns. Existing algorithms for CMTF are usually only able to handle exact couplings and a few special constraints. Gradient-based all-at-once optimization methods proposed for CMTF [10] can only handle box constraints. Other regularization and linear couplings are possible, but require the algorithm to be redesigned. The same holds for Gauss-Newton type methods as implemented in Tensorlab [13], where constraints are often realized via a transformation of variables. On the other hand, Huang et al. [14] proposed a flexible and efficient framework for constrained matrix and tensor factorizations that builds on Alternating Optimization (AO), where each subproblem is solved inexactly using the Alternating Direction Method of Multipliers (ADMM). It incorporates a wide variety of constraints and regularizations. However, coupled factorizations are not considered.

In this paper, we propose an optimization framework for constrained linearly coupled matrix-tensor factorizations that builds onto the AO-ADMM framework [14]. Experiments on synthetic data show that our flexible algorithmic approach

is accurate and achieves competitive performance compared to existing methods. We formulate the general optimization problem in Section II, and derive our algorithmic framework in Section III. Finally, we present the experiments in Section IV and conclude in Section V.

II. REGULARIZED LINEARLY COUPLED MATRIX AND TENSOR FACTORIZATION

We start by formulating the factorization problems considered in this paper in a general way. We consider N tensors or matrices $\{\mathcal{T}_i\}_{i=1,\dots,N}$, of not necessarily equal order $D_i \geq 2$ and size $n_{i,1} \times n_{i,2} \times \dots \times n_{i,D_i}$. We also suppose that each tensor \mathcal{T}_i follows approximately a CPD model with rank R_i ,

$$\mathcal{T}_i \approx \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i},$$

where $\mathbf{C}_{i,d} \in \mathbb{R}^{n_{i,d}, R_i}$ denotes the factor matrix of mode d in tensor \mathcal{T}_i . Moreover, we suppose that some factors $\mathbf{C}_{i,d}$ are regularized using proper convex lower semi-continuous functions $g_{i,d}(\mathbf{C}_{i,d})$. This covers the important case of constrained factors: suppose factor $\mathbf{C}_{i,d}$ should belong to a convex set $\mathcal{C}_{i,d}$, then we may set $g_{i,d} = \iota_{\mathcal{C}_{i,d}}$, where $\iota_{\mathcal{C}_{i,d}}$ is the indicator function that is null on $\mathcal{C}_{i,d}$ and infinity elsewhere. Also, $g_{i,d}$ can be a sparsity inducing norm such as the ℓ_1 norm. Finally, we suppose that some factors are shared across tensors. We consider the case of exact linearly coupled factors, where tensors \mathcal{T}_i may be coupled in any mode d via some underlying matrix $\Delta_d \in \mathbb{R}^{m_{1,d} \times m_{2,d}}$, as follows:

$$\mathbf{H}_{i,d} \text{vec}(\mathbf{C}_{i,d}) = \mathbf{H}_{i,d}^\Delta \text{vec}(\Delta_d), \quad i = 1, \dots, N,$$

for given transformation matrices $\mathbf{H}_{i,d} \in \mathbb{R}^{h_{i,d} \times R_i n_{i,d}}$ and $\mathbf{H}_{i,d}^\Delta \in \mathbb{R}^{h_{i,d} \times m_{1,d} m_{2,d}}$. $\text{vec}(\cdot)$ refers to columnwise vectorization. We illustrate some instances of such couplings in Section III-D. Using the squared Frobenius norm as a loss function, we aim at solving optimization problems of the form

$$\begin{aligned} & \underset{\{\mathbf{C}_{i,d}, \Delta_d\}_{d \leq D_i, i \leq N}}{\text{argmin}} && \sum_{i=1}^N w_i \left\| \mathcal{T}_i - \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i} \right\|_F^2 + \sum_{d=1}^{D_i} g_{i,d}(\mathbf{C}_{i,d}) \\ & \text{s.t.} && \mathbf{H}_{i,d} \text{vec}(\mathbf{C}_{i,d}) = \mathbf{H}_{i,d}^\Delta \text{vec}(\Delta_d) \end{aligned} \quad (1)$$

where w_i are weighting parameters.

III. ALGORITHMIC FRAMEWORK

A. Alternating Optimization

Problem (1) is non-convex and difficult to solve for all i and d simultaneously, in particular, when various constraints or regularizations are imposed on the factors. However, the problem w.r.t. the subset $\{\{\mathbf{C}_{i,d}\}_{i=1}^N, \Delta_d\}$ with fixed mode d , e.g., for mode 1,

$$\begin{aligned} & \underset{\{\mathbf{C}_{i,1}\}_{i \leq N}, \Delta_1}{\text{argmin}} && \sum_{i=1}^N w_i \left\| \mathcal{T}_i - \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i} \right\|_F^2 + g_{i,1}(\mathbf{C}_{i,1}) \\ & \text{s.t.} && \mathbf{H}_{i,1} \text{vec}(\mathbf{C}_{i,1}) = \mathbf{H}_{i,1}^\Delta \text{vec}(\Delta_1) \end{aligned} \quad (2)$$

is convex. A simple yet powerful idea is to partially solve (1) w.r.t. $\{\{\mathbf{C}_{i,1}\}_{i=1}^N, \Delta_1\}$ while keeping the factor matrices of modes $d = 2, \dots, N$ fixed, then w.r.t. $\{\{\mathbf{C}_{i,2}\}_{i=1}^N, \Delta_2\}$

and so on, and to iterate this process until convergence. This approach is called Alternating Optimization, or Block-coordinate descent, and is usually applied, when each sub-problem can be solved efficiently. As shown by Huang et al. [14] for the uncoupled problem, this can be achieved by using ADMM to (approximately) solve the convex sub-problems. The resulting optimization algorithm has been coined as AO-ADMM [14]. To the best of our knowledge, the AO-ADMM algorithm has no convergence guarantees, see also [14]. We derive the ADMM algorithm for the coupled sub-problem in the next subsection.

B. ADMM

ADMM is a primal-dual algorithm that aims at solving convex constrained optimization problems which can be written in the following form [15]:

$$\underset{\substack{\mathbf{x}, \mathbf{z} \\ \mathbf{Ax} + \mathbf{Bz} = \mathbf{c}}}{\text{argmin}} f(\mathbf{x}) + g(\mathbf{z}). \quad (3)$$

The scaled-form ADMM algorithm is given in Algorithm 1, where ρ is a parameter and $\boldsymbol{\mu}$ is a dual variable that corresponds to the constraint $\mathbf{Ax} + \mathbf{Bz} = \mathbf{c}$.

Algorithm 1 Skeleton of scaled-form ADMM [15]

```

while convergence criterion is not met do
   $\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\text{argmin}} f(\mathbf{x}) + \frac{\rho}{2} \left\| \mathbf{Ax} + \mathbf{Bz}^{(k)} - \mathbf{c} + \boldsymbol{\mu}^{(k)} \right\|_2^2$ 
   $\mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\text{argmin}} g(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{Ax}^{(k+1)} + \mathbf{Bz} - \mathbf{c} + \boldsymbol{\mu}^{(k)} \right\|_2^2$ 
   $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \mathbf{Ax}^{(k+1)} + \mathbf{Bz}^{(k+1)} - \mathbf{c}$ 
   $k = k + 1$ 
end while

```

C. ADMM for regularized CMTF with linear couplings

Without loss of generality, let us consider sub-problem (2) for mode 1. For each factor $\mathbf{C}_{i,1}$ we define “split” matrix variables $\mathbf{Z}_{i,1}$, similar to \mathbf{z} in (3), which separates the regularization from the factorization, but introduces the additional equality constraints $\mathbf{C}_{i,1} = \mathbf{Z}_{i,1}$. Variable Δ_1 can also be seen as a split variable since it decouples the coupled factor matrices of different tensors. This leads to an optimization problem directly expressed in the form of (3),

$$\begin{aligned} & \underset{\{\mathbf{C}_{i,1}, \mathbf{Z}_{i,1}\}_{i \leq N}, \Delta_1}{\text{argmin}} && \sum_{i=1}^N w_i \left\| \mathcal{T}_i - \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i} \right\|_F^2 + g_{i,1}(\mathbf{Z}_{i,1}) \\ & \text{s.t.} && \mathbf{H}_{i,1} \text{vec}(\mathbf{C}_{i,1}) = \mathbf{H}_{i,1}^\Delta \text{vec}(\Delta_1) \\ & && \mathbf{C}_{i,1} = \mathbf{Z}_{i,1} \end{aligned} \quad (4)$$

which can thus be solved with ADMM. In contrast to (3) and [14], we use ADMM with three blocks instead of two. Therefore, we introduce two sets of dual variables, $\boldsymbol{\mu}_{i,1(z)}$ for the regularization constraint and $\boldsymbol{\mu}_{i,1(\delta)}$ for the coupling constraint. The ADMM algorithm for mode 1 is given in Algorithm 2, where δ denotes $\text{vec}(\Delta)$. $\mathcal{T}_{[1]}$ denotes the mode-1 unfolding of tensor \mathcal{T} and we use the following short notation for Khatri-Rao products [9],

$$\mathbf{M}_{i,j} := \mathbf{C}_{i,D_i} \odot \dots \odot \mathbf{C}_{i,j+1} \odot \mathbf{C}_{i,j-1} \odot \dots \odot \mathbf{C}_{i,1}. \quad (5)$$

Algorithm 2 ADMM for regularized linearly coupled CPD

while convergence criterion is not met **do**
 for $i = 1, \dots, N$ **do**
 $\mathbf{C}_{i,1}^{(k+1)} = \underset{\mathbf{X}}{\operatorname{argmin}} w_i \left\| \mathcal{T}_{i[1]} - \mathbf{X} \mathbf{M}_{i,1}^T \right\|_F^2$
 $\quad + \frac{\rho}{2} \left\| \mathbf{X} - \mathbf{Z}_{i,1}^{(k)} + \boldsymbol{\mu}_{i,1(z)}^{(k)} \right\|_F^2$
 $\quad + \frac{\rho}{2} \left\| \mathbf{H}_{i,1} \operatorname{vec}(\mathbf{X}) - \mathbf{H}_{i,1}^\Delta \boldsymbol{\delta}_1^{(k)} + \boldsymbol{\mu}_{i,1(\delta)}^{(k)} \right\|_2^2$
 end for
 $\boldsymbol{\delta}_1^{(k+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} \sum_{i=1}^N \left\| \mathbf{H}_{i,1} \operatorname{vec}(\mathbf{C}_{i,1}^{(k+1)}) - \mathbf{H}_{i,1}^\Delta \mathbf{z} + \boldsymbol{\mu}_{i,1(\delta)}^{(k)} \right\|_2^2$
 for $i = 1, \dots, N$ **do**
 $\mathbf{Z}_{i,1}^{(k+1)} = \operatorname{prox}_{\rho, g_{i,1}} \left(\mathbf{C}_{i,1}^{(k+1)} + \boldsymbol{\mu}_{i,1(z)}^{(k)} \right)$
 $\boldsymbol{\mu}_{i,1(z)}^{(k+1)} = \boldsymbol{\mu}_{i,1(z)}^{(k)} + \mathbf{C}_{i,1}^{(k+1)} - \mathbf{Z}_{i,1}^{(k+1)}$
 $\boldsymbol{\mu}_{i,1(\delta)}^{(k+1)} = \boldsymbol{\mu}_{i,1(\delta)}^{(k)} + \mathbf{H}_{i,1} \operatorname{vec}(\mathbf{C}_{i,1}^{(k+1)}) - \mathbf{H}_{i,1}^\Delta \boldsymbol{\delta}_1^{(k+1)}$
 end for
 $k = k + 1$
end while

Note that both *for* loops can be computed using parallel programming. Furthermore, for uncoupled and unconstrained factor matrices, ADMM iterations are not necessary, since the exact solution can be obtained by a least squares update. The main advantage of ADMM is that it splits the problem (4) into easier individual problems for $\mathbf{C}_{i,1}$, $\mathbf{Z}_{i,1}$ and $\boldsymbol{\delta}_1$, respectively. As such, handling the regularization reduces to the computation of a proximal operator to update $\mathbf{Z}_{i,1}^{(k+1)}$. For any $\lambda > 0$, the proximal operator of g is the following function [16]

$$\operatorname{prox}_{\lambda, g}(\mathbf{x}) = \underset{\mathbf{u}}{\operatorname{argmin}} g(\mathbf{u}) + \frac{1}{2\lambda} \left\| \mathbf{x} - \mathbf{u} \right\|_2^2,$$

which is single-valued. For many functions g such as indicator functions of convex sets, a closed-form expression and/or an efficient implementation for the proximal operator is available, see lists here [16], [17].

D. Special cases of linear couplings

The first instruction in Algorithm 2 is a least squares problem, which has a closed form solution. However, it involves the implicit inversion of a large matrix, which can hardly be computed efficiently. Yet, provided that $\mathbf{H}_{i,1} \operatorname{vec}(\mathbf{C}_{i,1}) = \mathbf{H}_{i,1}^\Delta \operatorname{vec}(\boldsymbol{\Delta}_1)$ has a simpler form, it can be solved more efficiently. We differentiate between five possible forms of such easier linear couplings.

a) Case 1: Hard coupling (no transformation): One way of coupling tensors i and j is to require equality of factor matrices $\mathbf{C}_{i,1} = \mathbf{C}_{j,1}$. This case has been used in many CMTF formulations [1], [10], [11], [18]. The coupling constraint can then be written in matrix form as $\mathbf{C}_{i,1} = \boldsymbol{\Delta}_1$.

b) Case 2: Transformations in mode dimension: Often, measurements obtained from different instruments will correspond to different temporal or spatial sampling grids. For instance, suppose tensors i and j have different dimensions $n_{i,1}$, $n_{j,1}$ in mode 1, due to different sampling rates. It may still be possible to approximate the common underlying

function via interpolations on a common sampling grid of size n_{Δ_1} [12], where the variable $\boldsymbol{\Delta}_1 \in \mathbb{R}^{n_{\Delta_1} \times R}$ represents the function on the common grid. Two possibilities for such couplings are:

$$\begin{aligned} \text{a)} \quad & \tilde{\mathbf{H}}_{i,1} \mathbf{C}_{i,1} = \boldsymbol{\Delta}_1, & \tilde{\mathbf{H}}_{i,1} & \in \mathbb{R}^{n_{\Delta_1} \times n_{i,1}} \\ \text{b)} \quad & \mathbf{C}_{i,1} = \tilde{\mathbf{H}}_{i,1}^\Delta \boldsymbol{\Delta}_1, & \tilde{\mathbf{H}}_{i,1}^\Delta & \in \mathbb{R}^{n_{i,1} \times n_{\Delta_1}} \end{aligned}$$

To obtain a), set $\mathbf{H}_{i,1}^\Delta = \mathbf{I}_{R n_{\Delta_1}}$ and $\mathbf{H}_{i,1} = \mathbf{I}_R \otimes \tilde{\mathbf{H}}_{i,1}$. For b), set $\mathbf{H}_{i,1} = \mathbf{I}_{R n_{i,1}}$, $\mathbf{H}_{i,1}^\Delta = \mathbf{I}_R \otimes \tilde{\mathbf{H}}_{i,1}^\Delta$.

c) Case 3: Transformation in factor dimension: The following couplings allow tensors with the same size $n_{i,1} = n_{j,1} = n_1$ in mode 1, but different number of factors R_i , R_j , to be coupled,

$$\begin{aligned} \text{a)} \quad & \mathbf{C}_{i,1} \hat{\mathbf{H}}_{i,1} = \boldsymbol{\Delta}, & \hat{\mathbf{H}}_{i,1} & \in \mathbb{R}^{R_i \times R_\Delta} \\ \text{b)} \quad & \mathbf{C}_{i,1} = \boldsymbol{\Delta}_1 \hat{\mathbf{H}}_{i,1}^\Delta, & \hat{\mathbf{H}}_{i,1}^\Delta & \in \mathbb{R}^{R_\Delta \times R_i} \end{aligned}$$

To obtain a), set $\mathbf{H}_{i,1}^\Delta = \mathbf{I}_{R_\Delta n_1}$ and $\mathbf{H}_{i,1} = \hat{\mathbf{H}}_{i,1}^T \otimes \mathbf{I}_{n_1}$. For b), set $\mathbf{H}_{i,1} = \mathbf{I}_{R_i n_1}$ and $\mathbf{H}_{i,1}^\Delta = \hat{\mathbf{H}}_{i,1}^T \otimes \mathbf{I}_{n_1}$. In particular, information about shared and unshared factors can be encoded in rectangular “identity” matrices $\hat{\mathbf{H}}_i$. In Case 3a, the variable $\boldsymbol{\Delta}_1 \in \mathbb{R}^{n_{i,1} \times R_\Delta}$ contains R_Δ shared vectors of the possible larger factor matrices $\{\mathbf{C}_{i,1}\}_i$, while in Case 3b, $\boldsymbol{\Delta}_1$ is a wide matrix which contains all columns of all coupled factor matrices $\mathbf{C}_{i,1}$, no matter if they are shared or unshared. This is useful when some factors are only shared between some but not all coupled tensors. In Case 2a, a Sylvester equation has to be solved to update $\mathbf{C}_{i,1}$. In all other cases, the update of $\mathbf{C}_{i,1}$ in Algorithm 2 reduces to a linear least squares problem where the matrix inverse is only of size $R_i \times R_i$. Note that $\mathcal{T}_{i[1]} \mathbf{M}_{i,1}$ as well as $\mathbf{M}_{i,1}^T \mathbf{M}_{i,1}$ can be computed efficiently [9]. Furthermore, $\mathcal{T}_{i[1]} \mathbf{M}_{i,1}$ and the Cholesky decomposition of $(w_i \mathbf{M}_{i,1}^T \mathbf{M}_{i,1} + \rho \mathbf{I}_{R_i})$ can be precomputed outside the ADMM loop. The update of $\boldsymbol{\Delta}$ in Cases 1, 2a, 3a is simply given by an average, while in Cases 2b and 3b, a linear least squares problem has to be solved.

E. Algorithm details

We adapt the stopping criterion given in [15] for inner ADMM iterations. Algorithm 2 terminates when relative primal and dual residuals are below a threshold or when a predefined maximum number of inner iterations is reached. We empirically found that a threshold of 10^{-3} and 5 inner iterations work well for most problems. The whole algorithm terminates, when each of the following residuals,

$$\begin{aligned} f_{\text{tensors}}^{(k)} &= \sum_{i=1}^N w_i \left\| \mathcal{T}_i - \llbracket \mathbf{C}_{i,d} \rrbracket_{d=1}^{D_i} \right\|_F^2 \\ f_{\text{couplings}}^{(k)} &= \sum_{i,d} \left\| \mathbf{H}_{i,d} \operatorname{vec}(\mathbf{C}_{i,d}^{(k)}) - \mathbf{H}_{i,d}^\Delta \boldsymbol{\delta}_d^{(k)} \right\|_2 / \left\| \mathbf{H}_{i,d} \operatorname{vec}(\mathbf{C}_{i,d}^{(k)}) \right\| \\ f_{\text{constraints}}^{(k)} &= \sum_{i,d} \left\| \mathbf{C}_{i,d}^{(k)} - \mathbf{Z}_{i,d}^{(k)} \right\|_F / \left\| \mathbf{C}_{i,d}^{(k)} \right\|, \end{aligned} \tag{6}$$

has reached an absolute tolerance, has not changed more than some small relative tolerance, or a predefined number of maximal outer iterations is reached. For ρ , we use an adaptive

step-length $\rho_{i,d}^{(k)}$ [14] for the update of each factor matrix $\mathbf{C}_{i,d}^{(k)}$, where $\rho_{i,d}^{(k)} = \|\mathbf{M}_{i,d}^{(k)}\|_F^2 / R_i$.

IV. EXPERIMENTS

In this section, we assess the performance of the proposed AO-ADMM in terms of accuracy and computational efficiency on synthetic datasets, in comparison with state-of-the-art methods, namely Alternating Least Squares (ALS) and all-at-once optimization using quasi-Newton and Gauss-Newton.

A. Experimental Set-up

For ALS (referred to as CMTF-ALS), we use our own implementation for CMTF based on `cp_als` from the Tensor Toolbox [19]. For nonnegativity constraints, we solve the alternating nonnegative least squares problem using Hierarchical ALS (HALS) [20]. For all-at-once optimization (referred to as CMTF opt), we use `cmtf_opt` from the CMTF Toolbox [10] using Nonlinear Conjugate Gradient (NCG) for unconstrained cases and Limited Memory BFGS with bounds (LBFGS-B) for nonnegativity constraints. Finally, for Gauss-Newton (referred to as Tensorlab GN), we use the Tensorlab implementation `sdf_nls` [21], which can handle a variety of constraints and coupling structures. We monitor the convergence of different algorithms through the function value f_{tensors} and factor match score (FMS). Given the true factor matrices $\mathbf{C}_{i,d}^{\text{true}}$, the FMS for $\mathbf{C}_{i,d}$ is computed as

$$\text{FMS} = \prod_{i=1}^N \frac{1}{R_i} \sum_{r=1}^{R_i} \left(\prod_{d=1}^{D_i} \frac{\langle \mathbf{C}_{i,d}(:,r), \mathbf{C}_{i,d}^{\text{true}}(:,r) \rangle}{\|\mathbf{C}_{i,d}(:,r)\|_2 \|\mathbf{C}_{i,d}^{\text{true}}(:,r)\|_2} \right),$$

after finding the best permutation of factors. We run each algorithm until the relative change in function value is less than 10^{-12} or a maximum number of 10,000 outer iterations has been reached¹. For AO-ADMM, we set the absolute tolerance for the residuals in Eq. (6) to 10^{-4} .

For each experiment, we generated 50 random datasets. Following the CP model, tensors \mathcal{X}_i are constructed from known factor matrices. Tensors \mathcal{T}_i are then generated by adding noise tensors \mathcal{N}_i with entries drawn from a standard normal distribution as follows:

$$\mathcal{T}_i = \mathcal{X}_i + 0.2(\|\mathcal{X}_i\|_F / \|\mathcal{N}_i\|_F)\mathcal{N}_i.$$

This corresponds to a signal-to-noise ratio of around 14dB. We normalize each tensor and set the weights to $w_i = 1/2$.

For each dataset, multiple initializations (five for first three experiments and ten for the last - more difficult experiment) are used, and the best run with the lowest final function value is reported. We also report the number of failed runs in Table I, where a run is considered a failed run if it reaches the maximum number of iterations or gives an FMS below the threshold $0.99 \sum_{i=1}^N D_i$. When no nonnegativity constraints are imposed, factor matrices $\mathbf{C}_{i,d}$ are initialized using the first

¹Other parameters are set as follows: In `cmtf_opt` (NCG/LBFGS-B): `MaxFuncEvals/maxTotalIts=105`, `StopTol/pgtol=10-32`, in Tensorlab: `CGMaxIter=15`, `TolX=10-32`, `TolAbs=0`, in HALS: `maxiter=500`.

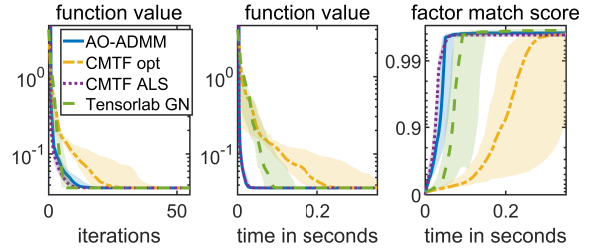


Fig. 1: Experiment 1: Median and quartiles of function values and FMS for different algorithms.

R left singular vectors of the corresponding concatenated (if coupled) and unfolded tensors in mode d in the first run. Otherwise, factor matrices are initialized at random, drawing from the standard normal or, in the case of nonnegativity, uniform distribution. All dual variables, as well as coupling variables Δ_d , are initialized using the standard normal distribution. The split variables $\mathbf{Z}_{i,d}$ are initialized by $\text{prox}_{1,g_{i,d}}(\mathbf{C}_{i,d})$.

B. Experiment 1: Case 1 and unconstrained

Here, we have an unconstrained coupled problem, where a third-order tensor of size $40 \times 50 \times 60$ is coupled using hard coupling in the first mode with a matrix of size 40×100 . We generate ground-truth factor matrices with $R = 3$ with entries randomly drawn from a normal distribution and transform them as described in [22], such that factors have a congruence of 0.5. Fig. 1 shows a summary of the convergence behaviour of different algorithms for 50 random datasets. Alternating methods CMTF-ALS and AO-ADMM behave similarly as we expect. All methods can also accurately recover the true factor matrices by achieving a FMS of close to 1. Tensorlab GN is able to achieve a better FMS near convergence. The initialization based on singular vectors seems to be a bad choice for AO-ADMM in the presence of collinearity. Table I shows a high number of failed runs for AO-ADMM which are all due to this type of initialization.

C. Experiment 2: Case 1 with nonnegativity constraints

The setting of the second experiment is similar to the first one. In addition, nonnegativity constraints are imposed on factor matrices in all modes. The ground-truth factor matrices are drawn from a uniform distribution and thus the collinearity of the components is not controlled. The convergence behaviour of different methods in Fig. 2 demonstrates that AO-ADMM is still computationally efficient, with an average performance similar to CMTF-ALS. All methods can also accurately recover the true factors used to generate the data.

D. Experiment 3: Case 2a and unconstrained

In this experiment, a tensor \mathcal{T}_1 of size $80 \times 50 \times 60$ is coupled in the first mode with a matrix \mathbf{T}_2 of size 40×100 via a transformation $\tilde{\mathbf{H}}_{1,1}\mathbf{C}_{1,1} = \Delta_1$, that discards every second entry in $\mathbf{C}_{1,1}$. True factor matrices with $R = 3$ have entries drawn from the standard normal distribution. We compare the AO-ADMM-based approach only with Tensorlab GN since no other implementation can currently handle linear couplings.

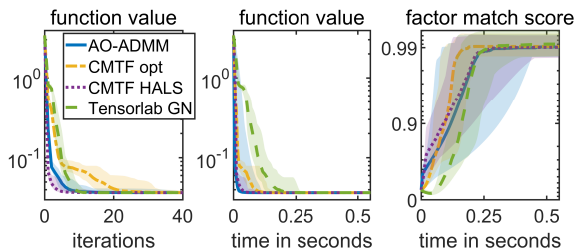


Fig. 2: Experiment2: Median and minimum/maximum function values and FMS for different algorithms.

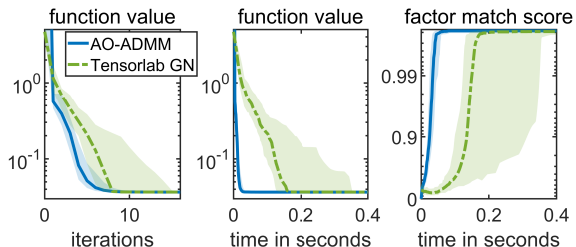


Fig. 3: Experiment3: Median and quartiles of function values and FMS for AO-ADMM and Tensorlab GN.

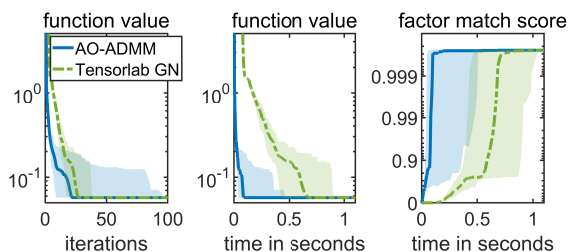


Fig. 4: Experiment4: Median and quartiles of function values and factor match scores for AO-ADMM and Tensorlab GN.

Fig. 3 shows that AO-ADMM finds the true factors faster than Tensorlab GN. Both are accurate while AO-ADMM seems less sensitive to initialization, see Table I.

E. Experiment 4: Case 3b and unconstrained

Here, three tensors of size $40 \times 50 \times 60$, $40 \times 70 \times 60$, $40 \times 30 \times 50$ and number of components $R = 2, 3, 4$, respectively, are coupled in the first mode. Two components are shared by all tensors while the additional component in the second tensor is also present in the third tensor. This is a more difficult problem with high number of failing runs as shown in Table I. However, when the best runs out of multiple initializations are considered, both AO-ADMM and Tensorlab GN are accurate while AO-ADMM finds the true factors faster, see Fig. 4.

V. CONCLUSIONS

We proposed a flexible algorithmic framework for regularized linearly coupled tensor factorizations using AO-ADMM, which can easily incorporate many constraints. The only requirement is that the proximal operator of the regularization function is tractable. Our experiments indicate that, while for hard coupled cases, AO-ADMM achieves comparable performance to other methods, it shows superior performance for linearly coupled problems. Nevertheless, more experiments with other constraints as well as more difficult datasets and

TABLE I. Failed runs

Exp.	out of	AO-ADMM	CMTF opt	CMTF ALS	TL GN
1	all/ best	45/0	0/0	6/0	12/0
2	all/ best	0/0	0/0	0/0	0/0
3	all/ best	4/0	-	-	56/0
4	all/ best	330/1	-	-	351/0

couplings are necessary. An extension of this framework to other loss functions is under study.

REFERENCES

- [1] E. E. Papalexakis, T. M. Mitchell, N. D. Sidiropoulos, C. Faloutsos, P. P. Talukdar, and B. Murphy, "Turbo-SMT: Accelerating coupled sparse matrix-tensor factorizations by 200x," in *SDM'14*, 2014.
- [2] S. Van Eyndhoven, B. Hunyadi, L. De Lathauwer, and S. Van Huffel, "Flexible data fusion of EEG-fMRI: Revealing neural-hemodynamic coupling through structured matrix-tensor factorization," in *EU-SIPCO'17*, 2017, pp. 26–30.
- [3] C. Chatzichristos, M. Davies, J. Escudero, E. Kofidis, and S. Theodoridis, "Fusion of EEG and fMRI via soft coupled tensor decompositions," *EUSIPCO'18*, pp. 56–60, 2018.
- [4] E. Acar, C. Schenker, Y. Levin-Schwartz, V. Calhoun, and T. Adali, "Unraveling diagnostic biomarkers of schizophrenia through structure-revealing fusion of multi-modal neuroimaging data," *Frontiers in Neuroscience*, vol. 13, p. 416, 2019.
- [5] E. Acar, R. Bro, and A. K. Smilde, "Data fusion in metabolomics using coupled matrix and tensor factorizations," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1602–1620, 2015.
- [6] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, no. 1, pp. 164–189, 1927.
- [7] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.
- [8] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.
- [9] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Sep. 2009.
- [10] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," in *KDD 2011 Workshop on Mining and Learning with Graphs*, 2011.
- [11] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining and Knowledge Discovery*, vol. 29, pp. 203–236, 2015.
- [12] R. Cabral Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Trans. Sig. Proc.*, vol. 64, no. 18, pp. 4830–4844, Sep. 2016.
- [13] L. Sorber, M. Van Barel, and L. De Lathauwer, "Structured data fusion," *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 4, pp. 586–600, 2015.
- [14] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5052–5065, 2016.
- [15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan 2011.
- [16] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017.
- [17] <http://proximity-operator.net/proximityoperator.html>.
- [18] T. F. Wilderjans, E. Ceulemans, and I. Van Mechelen, "Simultaneous analysis of coupled data blocks differing in size: A comparison of two weighting schemes," *Comput. Stat. Data Anal.*, vol. 53, pp. 1086–1098, 2009.
- [19] B. W. Bader, T. G. Kolda *et al.*, "Matlab tensor toolbox version 3.1," Jun. 2019. [Online]. Available: <https://www.tensortoolbox.org>
- [20] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization," *Neural computation*, vol. 24, no. 4, pp. 1085–1105, 2012.
- [21] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v3. 0," available online, URL: www.tensorlab.net, 2016.
- [22] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the parafac model," *Comput. Stat. Data Anal.*, vol. 50, no. 7, pp. 1700 – 1734, 2006.